In the Claims:

1. An apparatus operating an N_logalu of a logalu, each using a wide instruction, comprising:

a narrow instruction provided to a local wide instruction memory;

said local wide instruction memory at least partly creating said wide instruction in response to said narrow instruction;

said logalu receiving said wide instruction;

said logalu configured based upon said wide instruction to operate upon N_log_operands of a log-operand;

wherein said N_logalu is at least one; and

wherein said N_log_operands is at least two.

2. The apparatus of Claim 1, wherein said N_logalu is at least two.

3. The apparatus of Claim 2, wherein said logalu receiving said wide instruction is further comprised, for each of said logalus:

said logalu receiving said wide instruction.

4. The apparatus of Claim 2, wherein said wide instruction for at least two of said logalus is at least partly distinct.

5. The apparatus of Claim 4, wherein said local wide instruction memory responding to said narrow instruction is further comprised of:

said local wide instruction memory responding to said narrow instruction to at least partly create said wide instruction associated with said logalu, for each of said logalus;

wherein said wide instructions associated with at least two of said logalus are at least partly distinct.

6. The apparatus of Claim 1, wherein said narrow instruction is comprised of a designator field, a first narrow field and a second narrow field;

wherein said local wide instruction memory responding to said narrow instruction, is further comprised of:

a means for selecting from said first narrow field and said second narrow field based upon said designator field to provide a selected narrow instruction to a local wide memory; and

said local wide memory responding to said selected narrow instruction to at least partly create said wide instruction.

7. The apparatus of Claim 6, wherein the means for selecting is further comprised of:

presenting said designator field to a selection configuration circuit; and

said selection configuration circuit responding to said designator field to select from said first narrow field and said second narrow field to create said selected narrow instruction.

8. The apparatus of Claim 7, further comprising:

means for providing a configuration signal to said selection configuration circuit; and

means for altering a state in said selection configuration circuit based upon said configuration signal.

9. The apparatus of Claim 8, wherein said state resides in a selection configuration memory included in said selection configuration circuit.

10. The apparatus of Claim 1, further comprising:

means for providing said local wide instruction memory with a write instruction;

wherein said local wide instruction memory responding to said narrow instruction is at least partly altered based upon said write instruction.

11. A apparatus of generating a non-additive result based upon said N_log_operands of a log-operand of Claim 1, comprising:

said apparatus of Claim 1 operating said logalu;

means for providing said log-operands to said logalu;

said logalu responding to said log-operands and to said wide instruction to create a log-result presented to an exponential calculator; and

said exponential calculator responding to said log-result to create said non-additive result.

12. The apparatus of Claim 11, wherein said non-additive result includes a first multiplicative

component and a second multiplicative component;

wherein said first multiplicative component is a member of the non-additive primitive collection based upon a first of said log-operands;

wherein said second multiplicative component is a member of said non-additive primitive collection based upon a second of said log-operands;

wherein said non-additive primitive collection based upon one of said log-operands includes:

an exponential of said one log-operand,

an exponential of a half of said one log-operand,

an exponential of a negative of said one log-operand,

an exponential of a negative of said half of said one log-operand,

an exponential of a double of said one log-operand, and

an exponential of a negative of said double of said one log-operand.

13. The apparatus of Claim 12, further comprising the step of:

a log-calculator providing said first log-operand based upon a first operand;

wherein said non-additive primitive collection based upon one of said log-operands includes:

an approximation of said first operand,

an approximation of a square root of said first operand,

an approximation of a multiplicative inverse of said first operand,

an approximation of a multiplicative inverse of said square root of said first operand,

an approximation of a square of said first operand, and

an approximation of a multiplicative inverse of said square of said first operand.

14. The apparatus of Claim 13, wherein said approximations satisfy a precision standard.

15. The apparatus of Claim 14, wherein said precision standard supports a member of the programming languages collection comprising: a version of Java, a version of C, a version of OpenGL, and a version of DirectX.

16. A method of operating an N_logalu of a logalu using a wide instruction, comprising the steps of:

providing a narrow instruction to a local wide instruction memory;

said local wide instruction memory responding to said narrow instruction to at least partly create said wide instruction;

said logalu receiving said wide instruction;

configuring said logalu based upon said wide instruction to operate upon an N_log_operands of a log-operand;

wherein said N_logalu is at least one; and

wherein said N_log_operands is at least two.

17. The method of Claim 16, wherein said N_logalu is at least two.

18. The method of Claim 17, wherein the step of said logalu receiving said wide instruction is further comprised, for each of said logalus, of the step of:

said logalu receiving said wide instruction.

19. The method of Claim 17, wherein said wide instruction for at least two of said logalus is at least partly distinct.

20. The method of Claim 18, wherein the step said local wide instruction memory responding to said narrow instruction is further comprised of the step of:

said local wide instruction memory responding to said narrow instruction to at least partly create said wide instruction associated with said logalu, for each of said logalus;

wherein said wide instructions associated with at least two of said logalus are at least partly distinct.

21. The method of Claim 16, wherein said narrow instruction is comprised of a designator field, a first narrow field and a second narrow field;

wherein the step said local wide instruction memory responding to said narrow instruction, is further comprised of the steps of:

selecting from said first narrow field and said second narrow field based upon said designator field to provide a selected narrow instruction to a local wide memory; and

said local wide memory responding to said selected narrow instruction to at least partly create

said wide instruction.

22. The method of Claim 21, wherein the step of selecting is further comprised of the steps of:

presenting said designator field to a selection configuration circuit; and

selecting from said first narrow field and said second narrow field based upon a response from said selection configuration circuit to said designator field.

23. The method of Claim 22, further comprising the steps of:

providing a configuration signal to said selection configuration circuit; and

altering a state in said selection configuration circuit based upon said configuration signal.

24. The method of Claim 23, wherein said state resides in a selection configuration memory included in said selection configuration circuit.

25. The method of Claim 16, further comprising the step of:

providing said local wide instruction memory with a write instruction;

wherein the step of said local wide instruction memory responding to said narrow instruction is at least partly altered based upon said write instruction.

26. A method of generating a non-additive result based upon said N_log_operands of a log-operand of Claim 16, comprising the steps of:

operating a logalu by the steps of Claim 16,

providing said log-operands to said logalu;

said logalu responding to said log-operands and to said wide instruction to create a log-result presented to a exponential calculator; and

said exponential calculator responding to said log-result to create said non-additive result.

27. The method of Claim 26, wherein said non-additive result includes a first multiplicative component and a second multiplicative component;

wherein said first multiplicative component is a member of the non-additive primitive collection based upon a first of said log-operands;

wherein said second multiplicative component is a member of said non-additive primitive collection based upon a second of said log-operands;

wherein said non-additive primitive collection based upon one of said log-operands includes:

an exponential of said one log-operand,

an exponential of a half of said one log-operand,

an exponential of a negative of said one log-operand,

an exponential of a negative of said half of said one log-operand,

an exponential of a double of said one log-operand, and

an exponential of a negative of said double of said one log-operand.

28. The method of Claim 27, further comprising the step of:

a log-calculator providing said first log-operand based upon a first operand;

wherein said non-additive primitive collection based upon one of said log-operands includes:

an approximation of said first operand,

an approximation of a square root of said first operand,

an approximation of a multiplicative inverse of said first operand,

an approximation of a multiplicative inverse of said square root of said first operand,

an approximation of a square of said first operand, and

an approximation of a multiplicative inverse of said square of said first operand.

29. The method of Claim 28, wherein said approximations satisfy a precision standard.

30. The method of Claim 29, wherein said precision standard supports a member of the programming languages collection comprising: a version of Java, a version of C, a version of OpenGL, and a version of DirectX.